



## **libParamTuner : interactive tuning of parameters without code recompilation**

Marc Baloup, Veis Oudjail, Géry Casiez

### **► To cite this version:**

Marc Baloup, Veis Oudjail, Géry Casiez. libParamTuner : interactive tuning of parameters without code recompilation. 29ème conférence francophone sur l'Interaction Homme-Machine, AFIHM, Aug 2017, Poitiers, France. 2 p. hal-01577686

**HAL Id: hal-01577686**

**<https://hal.science/hal-01577686>**

Submitted on 27 Aug 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# libParamTuner : réglage interactif de paramètres sans re-compilation de code

## *libParamTuner : interactive tuning of parameters without code recompilation*

Marc Baloup, Veis Oudjail et Géry Casiez

Université de Lille, France

{marc.baloup, veis.oudjail}@etudiant.univ-lille1.fr, gery.casiez@univ-lille1.fr

### ABSTRACT

We present libParamTuner, a cross-platform library designed to allow the interactive tuning of parameters in applications written in C++ or Java, without the need to re-compile code. libParamTuner provides a lightweight syntax to bind some variables of an application to the parameters defined in an XML file. Each modification of the XML file updates in real time the associated parameters in the application. A graphical interface allows editing the XML file, using interactive controls dynamically created for each parameter.

### CCS CONCEPTS

• **Human-centered computing** → **User interface toolkits**;

### KEYWORDS

library, parameters, interactive tuning.

### RÉSUMÉ

Nous présentons libParamTuner, une librairie multi-plateformes conçue pour permettre le réglage interactif de paramètres dans des applications écrites en C++ ou Java, sans avoir besoin de recompiler le code source. libParamTuner propose une syntaxe légère pour associer des variables d'une application à des paramètres définis dans un fichier XML. Toute modification du fichier XML entraîne une mise à jour en temps réel des variables pour les paramètres correspondants. Une interface graphique permet d'éditer le fichier XML, en utilisant des contrôles interactifs créés dynamiquement pour chaque paramètre.

### MOTS-CLEFS

librairie, paramètres, réglage interactif.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

IHM'17, August 28–September 1, 2017, Poitiers, France

### 1 INTRODUCTION

Le développement d'applications requiert souvent la définition d'un certain nombre de paramètres dont les valeurs par défaut sont difficiles à définir a priori. C'est particulièrement vrai pour les systèmes interactifs où, par exemple, la position d'un objet à l'écran ou sa vitesse de déplacement sont difficiles à définir précisément par avance. C'est également le cas pour le réglage de paramètres de filtres conçus pour des systèmes interactifs comme le 1€ filter [2]. Le réglage de chacun des paramètres de ce filtre demande en effet une appréciation par l'utilisateur de l'effet obtenu qui se fait plus facilement quand il est possible de régler interactivement chacun des paramètres.

Généralement le développeur utilise une stratégie essai/erreur : il donne une valeur plus ou moins arbitraire aux différents paramètres à régler, compile le code de l'application, exécute et observe le résultat obtenu. Il réitère l'opération jusqu'à obtenir un résultat satisfaisant, ce qui peut être long et fastidieux, sans garantie d'avoir obtenu le meilleur réglage possible. Le processus est d'autant plus difficile quand il est nécessaire de régler des paramètres qui sont corrélés.

Une alternative est de permettre la modification de ces paramètres dans l'interface, par exemple avec les touches du clavier. Cela demande d'écrire du code spécifique, ce qui prend du temps. Suivant le nombre de paramètres à régler, il peut être difficile de mémoriser les touches associées à chaque paramètre, et l'affichage des valeurs courantes des paramètres se fait le plus souvent dans la console. Un autre exemple est de développer une interface graphique dans l'application pour régler de manière interactive ces paramètres. Là encore le développement de l'interface graphique peut être couteux en temps. Ce coût est généralement jugé trop élevé par rapport aux bénéfices espérés.

Certains langages comme python en mode interactif ou des langages dérivés de Smalltalk comme Pharo [1] facilitent la modification de valeurs de paramètres à l'exécution. Cependant le contrôle interactif de ces paramètres via une interface graphique n'est pas prévu par défaut et les modifications ne sont pas enregistrées de manière persistante. Par ailleurs la majorité des applications développées aujourd'hui utilisent d'autres langages de programmation comme C++ ou Java.

D'autres approches comme Kaboom [3] ou Choc<sup>1</sup> s'inspirent d'idées popularisées par Bret Victor sur le *Leanable Programming*<sup>2 3</sup>

1. <https://www.fullstack.io/choc/>

2. <http://worrydream.com/LeanableProgramming/>

3. <https://vimeo.com/36579366>

pour modifier de mani re interactive des param tres dans des IDE particuliers et pour des langages sp cifiques. libParamTuner est inspir e de ces travaux mais se veut tr s facile   utiliser, sans avoir besoin de modifier l'environnement de d veloppement de l'utilisateur. Son fonctionnement est d taill  ci-dessous.

## 2 LIBPARAMTUNER

libParamTuner comprend trois parties : 1) un fichier texte au format XML (d finition des param tres) 2) une librairie C++ qui observe quand le fichier texte est modifi  et met   jour   la vol e les valeurs dans l'application 3) une application de r glage interactif des param tres qui charge le fichier, cr e les widgets de contr le des param tres et sauvegarde les modifications dans le fichier texte.

L'utilisation d'un fichier XML permet une d finition simple de chaque param tre et surtout une sauvegarde persistante des r glages effectu s. Les types bool en, entier, r el (float, double) et chaines de caract res sont support s.

Un exemple minimaliste en C++ est le suivant :

```
#include "paramtuner.h"
int main() {
    int i; double d; string s;
    ParamTuner::load("settings.xml");
    ParamTuner::bind("varI", &i);
    ParamTuner::bind("varD", &d);
    ParamTuner::bind("varS", &s);
    loop(); // Boucle principale du programme.
}
```

Le d veloppeur commence par d clarer les variables correspondantes aux param tres. Il d finit ensuite le chemin vers le fichier XML et associe chaque variable   un n ud du fichier XML, dont voici l'exemple associ  au code ci-dessus :

```
<?xml version="1.0" encoding="UTF-8"?>
<ParamList>
  <varD type="double" value="2.3" min="0" max="100"/>
  <varI type="int" value="12" min="0" max="100"/>
  <varS type="string" value="lorem ipsum"/>
</ParamList>
```

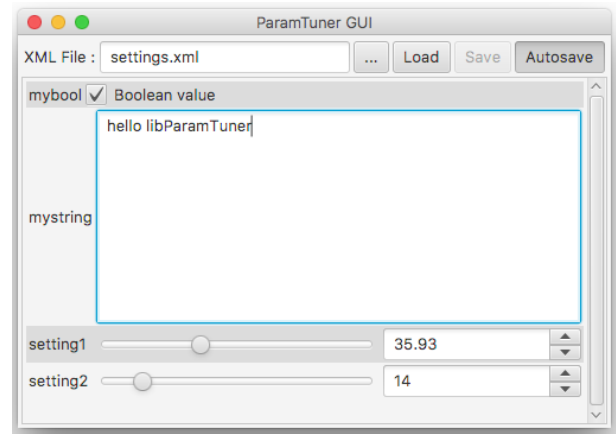
Chaque n ud comprend le nom du param tre, son type, sa valeur courante et, pour les types num riques, les valeurs minimales et maximales, utilis es par l'application de r glage interactif.

La surveillance des modifications du fichier XML se fait par l'utilisation de librairies natives sp cifiques   chaque syst me d'exploitation. Nous utilisons la fonction FindFirstChangeNotification sous Windows, la librairie inotify sous Linux et les FSEvents sous macOS.

Chaque modification de fichier d clenche l'appel d'une fonction de callback qui modifie les valeurs stock es aux adresses m moires des variables associ es   chaque param tre. Dans l'application, la seule contrainte est de forcer la mise   jour r guli re de l'affichage afin de lire les nouvelles valeurs des param tres.

## 3 R GLAGE INTERACTIF

Le r glage interactif des param tres se fait par l'utilisation d'une application tierce  crite en Java. Celle-ci permet de charger un



**FIGURE 1: L'application de r glage interactif des param tres. L'application permet de d finir le fichier XML   charger et g n re automatiquement les widgets permettant le r glage de chacun des param tres. Le fichier est soit sauvegard  automatiquement   chaque modification ou manuellement.**

fichier XML et g n re l'ensemble des widgets associ s   chacun des param tres : cases   cocher pour des bool ens, sliders et champs de texte pour les entiers et r els et champs de texte pour les chaines de caract res (Figure 1). La sauvegarde du fichier peut se faire manuellement ou automatiquement   chaque modification d'un param tre. En pratique le d lai de mise   jour des param tres est tr s faible (inf rieur   la seconde), m me si nous n'avons pas pu le caract riser formellement.

## 4 CONCLUSION

libParamTuner permet de r gler de mani re interactive les param tres d'une application sans avoir besoin de la recompiler. libParamTuner est librement disponible sur GitHub<sup>4</sup> sous licence GPL. La librairie peut  tre utilis e dans des programmes  crits en C++ ou Java mais nous esp rons que la communaut  aidera au d veloppement de bindings dans d'autres langages.

## R F RENCES

- [1] Alexandre Bergel, Damien Cassou, St phane Ducasse, and Jannik Laval. 2013. *Deep into Pharo*. Square Bracket Associates. 420 pages. <https://hal.inria.fr/hal-00858725>
- [2] G ry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1  Filter : A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *Proceedings of CHI '12, the 30th Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2527–2530. DOI : <http://dx.doi.org/10.1145/2207676.2208639>
- [3] Marchal Damien. 2014. Kaboom : une bo te   outils pour la programmation   la vol e en C/C++. IHM'14, 26e conf rence francophone sur l'Interaction Homme-Machine. (Oct. 2014). <https://hal.archives-ouvertes.fr/hal-01089615> Poster.

4. <https://github.com/casiez/libparamtuner>